Inheritance

• *Inheritance*: a mechanism for creating a new class from an existing one.

• *Motivation*:
  • Can *reuse* existing classes
    • Faster than writing them from scratch
public class Square extends Rectangle {
    private double diagonal;
    public Square(int side) {
        super(side, side); // superclass constructor
        diagonal = (double)side * 1.4142;
    }
    public int getSide() {
        return getWidth();
    }
    public String toString() {
        return "Square: Side(" + getSide() + ")";
    }
}

/** A class that models a rectangle */
public class Rectangle {
    private int length;
    private int width;

    public Rectangle(int len, int w) {
        length = len;
        width = w;
    }
    public int getLength() {
        return length;
    }
    public int getWidth() {
        return width;
    }

    public int area() {
        return length*width;
    }

    public String toString() {
        return "Rectangle: Length(" + length + "), Width(" + width + ");"
    }
}
public class Square extends Rectangle {
    private double diagonal;
    public Square(int side) {
        super(side, side);
        diagonal = (double)side * 1.4142;
    }

    public String toStringAsRectangle() {
        return toString();
    }

    public String toString() {
        return "Square: Side(" + getSide() + ")";
    }
}
Inheritance

Class A methods and instance variables

Class B methods and instance variables

Class A

Parent or base class

Class B

Child class or subclass
Inheritance

Class A methods and instance variables

Class B methods and instance variables

Class C methods and instance variables

Class A

Class B

Class C

Parent or base class

Child class or subclass